Forecasting and Risk (BANA 4090)

Forecasting Basics (Part I)

Zhaohu(Jonathan) Fan

06/08/2021

- Main topics:
    - Four simple forecasting models
    - Residual diagnostics

Prerequisites

```r
# List of required (CRAN) packages
pkgs <- c(
  "ggplot2",# for drawing nicer graphics
  "fpp2",    # for using four simple forecasting models
  "forecast", #for using checkresiduals() function: a test of autocorrelation of the
)

# Install required (CRAN) packages
for (pkg in pkgs) {
  if (!(pkg %in% installed.packages()[, "Package"])) {
    install.packages(pkg)
  }
}
```

Simple forecasting models

Forecasting is a risky business!

There are three distinct sources of forecasting risk:

- Intrinsic risk is random variation that is beyond explanation with the data and tools you have available. It's the "noise" in the system.
- Parameter risk is the risk due to errors in estimating the parameters of the forecasting model you are using.
- Model risk is the risk of choosing the wrong model.

Four simple forecasting models

Average model

- Forecast of all future values is equal to mean of historical data $\{y_1, \dots, y_T\}$.
- Forecasts: $\hat{y}_{T+h|T} = \bar{y} = (y_1 + \cdots + y_T)/T$

Naïve model

- Forecasts equal to last observed value.
- Forecasts: $\hat{y}_{T+h|T} = y_T$.
- Consequence of efficient market hypothesis.

Seasonal naïve model

- Forecasts equal to last value from same season.
- Forecasts: $\hat{y}_{T+h|T} = y_{T+h-m(k+1)}$, where $m =$ seasonal period and k is the integer part of $(h-1)/m$.

Four simple forecasting models

Drift model

- Forecasts equal to last value plus average change.
- Forecasts: $\hat{y}_{T+h|T} = y_T + \frac{h}{T-1}\sum_{t=2}^{T}(y_t - y_{t-1}) = y_T + \frac{h}{T-1}(y_T - y_1)$.
- Equivalent to extrapolating a line drawn between first and last observations.

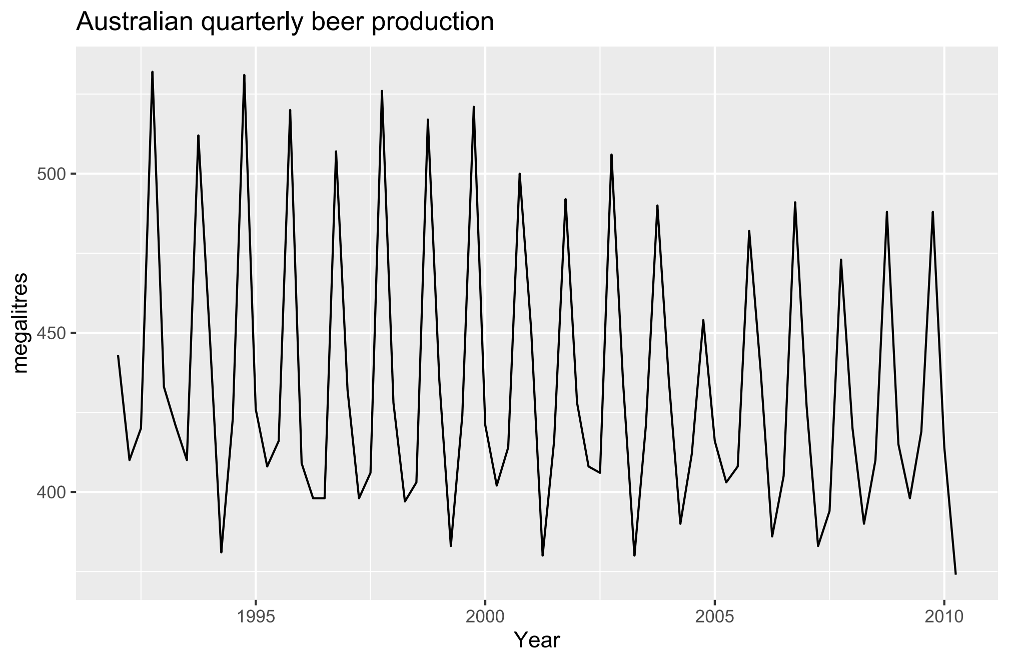
Four simple forecasting models in R

- Average (Mean) model: `meanf(y, h=20)`
- Naïve model: `naive(y, h=20)`
- Seasonal naïve model: `snaive(y, h=20)`
- Drift model: `rwf(y, drift=TRUE, h=20)`

Data example

- Total quarterly beer production in Australia (in megalitres) from 1956:Q1 to 2010:Q2.

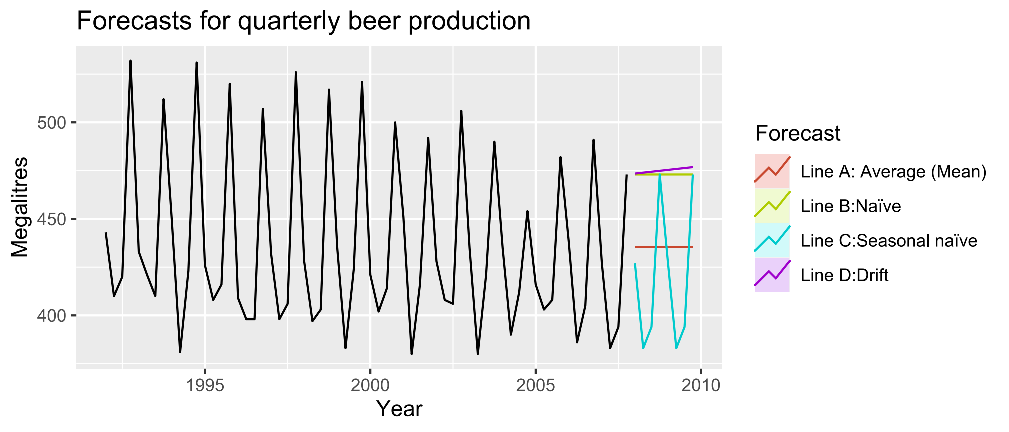
```
beer2 <- window(ausbeer, start=1992)
autoplot(beer2) +
  xlab("Year") + ylab("megalitres") +
    ggtitle("Australian quarterly beer production")
```



Data example (cont'd)

- Four simple forecasting models

```
beer2 <- window(ausbeer,start=1992,end=c(2007,4))
# Plot some forecasts
autoplot(beer2) +
  autolayer(meanf(beer2, h=8), PI=FALSE, series="Line A: Average (Mean)") +
  autolayer(naive(beer2, h=8), PI=FALSE, series="Line B:Naïve") +
  autolayer(snaive(beer2, h=8), PI=FALSE, series="Line C:Seasonal naïve") +
  autolayer(rwf(beer2, drift=TRUE, h=8), PI=FALSE, series="Line D:Drift") +
  ggtitle("Forecasts for quarterly beer production") +
  xlab("Year") + ylab("Megalitres") +
  guides(colour=guide_legend(title="Forecast"))
```

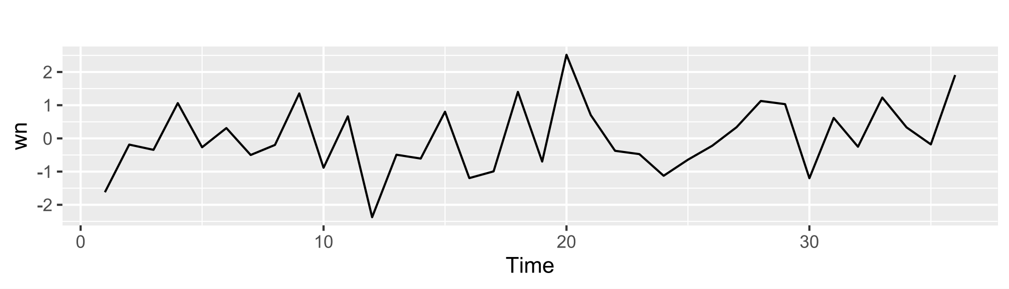


Forecasting

How do we know when our model is good, or at least not obviously bad?

- One very basic test of our model is whether its errors (residuals) really look like pure noise (white noise), i.e., independent and identically distributed random variables.
    - **Example: white noise**

```
wn ← ts(rnorm(36))
autoplot(wn)
```



- If the errors (residuals) are not pure noise (white noise), then by definition there are some patterns in them, and we could make them smaller by adjusting the model to explain that pattern.

Signal vs. Noise

- The **signal** is the predictable component in a forecasting model, and the **noise** is what is left over.
- The technical term for a data series that is pure noise is that it is a sequence of “independent and identically-distributed (i.i.d.) random variables.”

Residual diagnostics

Residual diagnostics

- Residuals in forecasting: difference between observed value and its fitted value: $e_t = y_t - \hat{y}_{t|t-1}$.

- Assumptions:
  - (1) $\{e_t\}$ are **uncorrelated**. If they aren't, then information left in residuals that should be used in computing forecasts.
  - (2) $\{e_t\}$ have **mean zero**. If they don't, then forecasts are biased.

- Useful properties (for prediction intervals):
  - (1) $\{e_t\}$ have **constant variance**.
  - (2) $\{e_t\}$ are **normally distributed**.

Autocorrelation (ACF)

Covariance and **correlation**: measure extent of **linear relationship** between two variables (Y and X).

Autocovariance and **autocorrelation**: measure linear relationship between **lagged values** of a time series y.

We measure the relationship between:

- y_t and y_{t-1}
- y_t and y_{t-2}
- y_t and y_{t-3}
- etc.

ACF of residuals

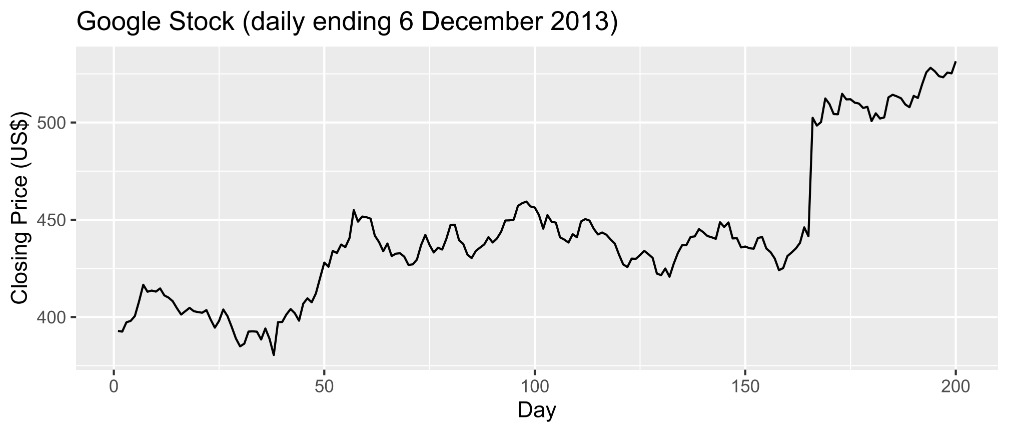
- We assume that the residuals are white noise (**uncorrelated**, mean zero, constant variance). If they aren't, then there is information left in the residuals that should be used in computing forecasts.

- So a standard residual diagnostic is to check the ACF of the residuals of a forecasting model.

- We expect these to look like white noise.

Data Example: Google stock price

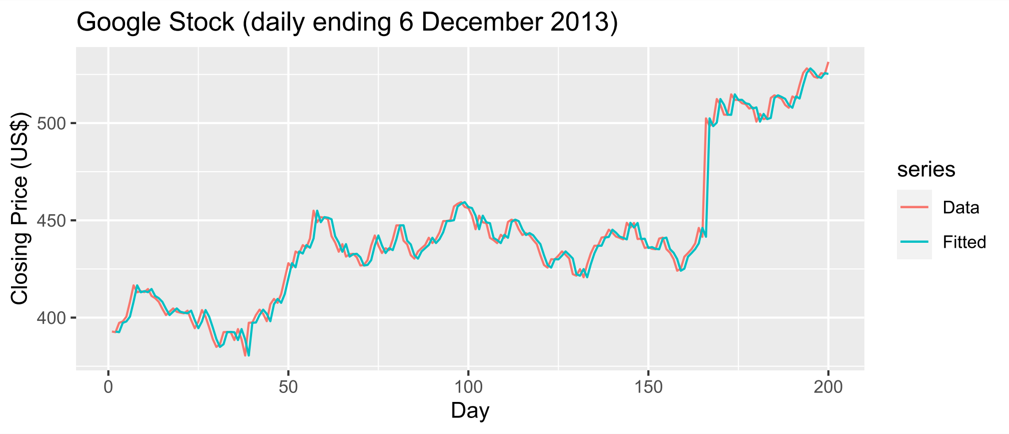
```
autoplot(goog200) +
  xlab("Day") + ylab("Closing Price (US$)") +
  ggtitle("Google Stock (daily ending 6 December 2013)")
```



Data example (cont'd)

- Naïve model

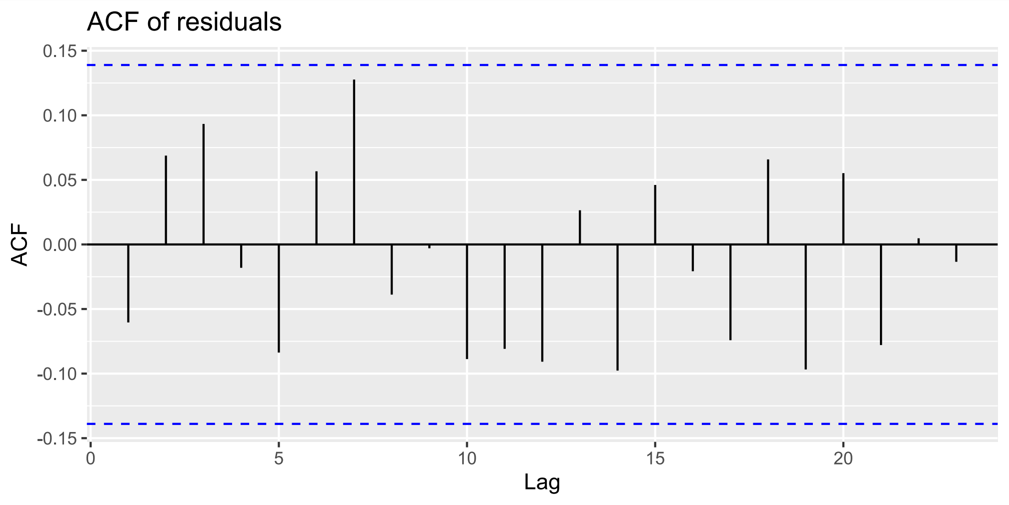
```
fits ← fitted(naive(goog200))
autoplot(goog200, series="Data") +
  autolayer(fits, series="Fitted") +
  xlab("Day") + ylab("Closing Price (US$)") +
  ggtitle("Google Stock (daily ending 6 December 2013)")
```



Data example (cont'd)

- check if the residuals are white noise (**uncorrelated?**).
- We expect each autocorrelation to be close to zero.

```
res ← residuals(naive(goog200))
ggAcf(res) + ggtitle("ACF of residuals")
```

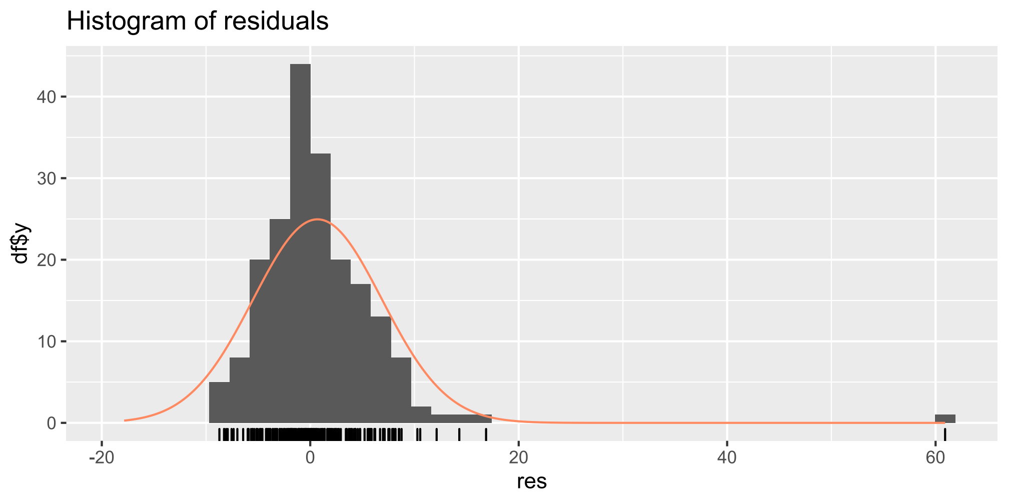


- There is no significant correlation in the residuals series.The vertical lines (lags) don't pass through horizontal blue lines. This means the autocorrelations are not (statistically) significantly different from zero.

Data example (cont'd)

- check if the residuals are white noise (**normally distributed+ mean zero?**).

```
gghistogram(res, add.normal=TRUE) +
  ggtitle("Histogram of residuals")
```

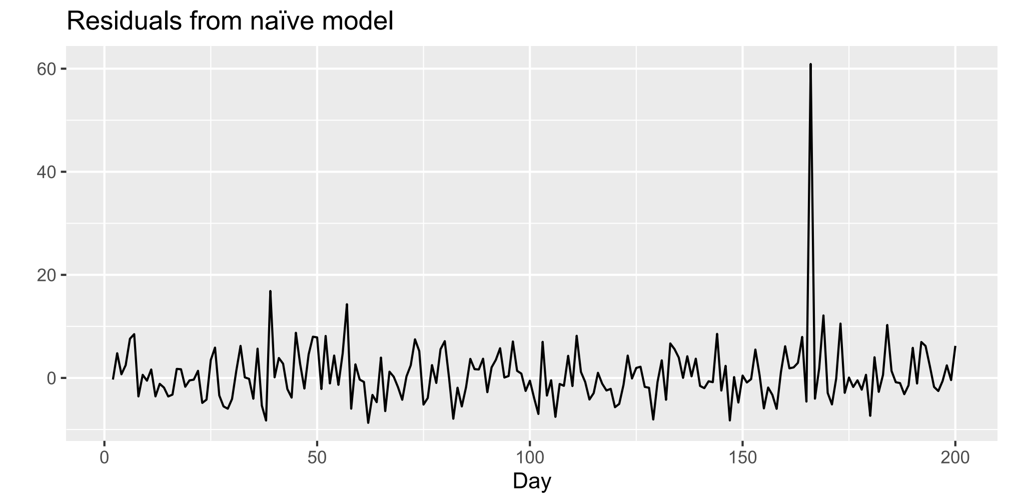


- The mean of the residuals is close to zero.
- The histogram suggests that the residuals may not be normal. The right tail seems a little too long, even when we ignore the outlier.

Data example (cont'd)

- check if the residuals are white noise (**constant variance?**).

```
autoplot(res) + xlab("Day") + ylab("") +
  ggtitle("Residuals from naïve model")
```



- The time plot of the residuals shows that the variation of the residuals stays much the same across the historical data, apart from the one outlier, and therefore the residual variance can be treated as constant.

Data example (cont'd): Conclusion

- Forecasts from this method (Naïve model) will probably be quite good,
- but prediction intervals that are computed assuming a normal distribution may be inaccurate.

Ljung-Box test

- In general, the Ljung-Box test (a test of autocorrelation of the residuals) is defined as:
  - H_0 (null hypothesis): the autocorrelations come from a white noise series.
  - H_a (alternative hypothesis): the autocorrelations do not come from a white noise series.

- Ljung-Box test is based on
  - $Q^* = T(T+2)\sum_{k=1}^{h}(T-k)^{-1}r_k^2$ where h is max lag being considered and T is number of observations.
  - large values of Q^* suggest that the autocorrelations do not come from a white noise series.

- By convention: $h = 10$ for non-seasonal data, $h = 2m$ for seasonal data, where m is the period of seasonality.

- Better performance, especially in small samples.

Ljung-Box test (cont'd)

- If data are white noise, Q^* has χ^2 distribution with $(h - K)$ degrees of freedom where $K =$ no. parameters in model.
- For the Google stock price example, the naïve model has no parameters, so $K = 0$

```
# lag=h and fitdf=K
# Total lags used: 10
Box.test(res, lag=10, fitdf=0, type="Lj")
```

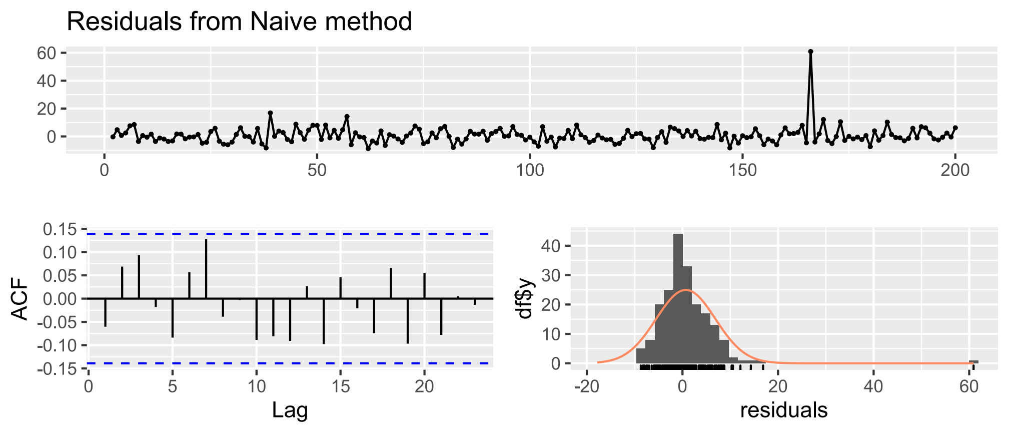
```
##
##      Box-Ljung test
##
## data:  res
## X-squared = 11.031, df = 10, p-value = 0.3551
```

- **The results are not statistically significant (i.e., the p-value>0.05). We fail to reject the null hypothesis H_0. Thus, we can conclude that the residuals are not distinguishable from a white noise series**.

Ljung-Box test (cont'd)

- Test if the residuals are white noise.
- `checkresiduals` function produces 1) a time plot of residuals, 2) a ACF of the residuals, 3) a histogram of the residuals (with an overlaid normal distribution for comparison), and 4) a Ljung-Box test.

```
checkresiduals(naive(goog200))
```

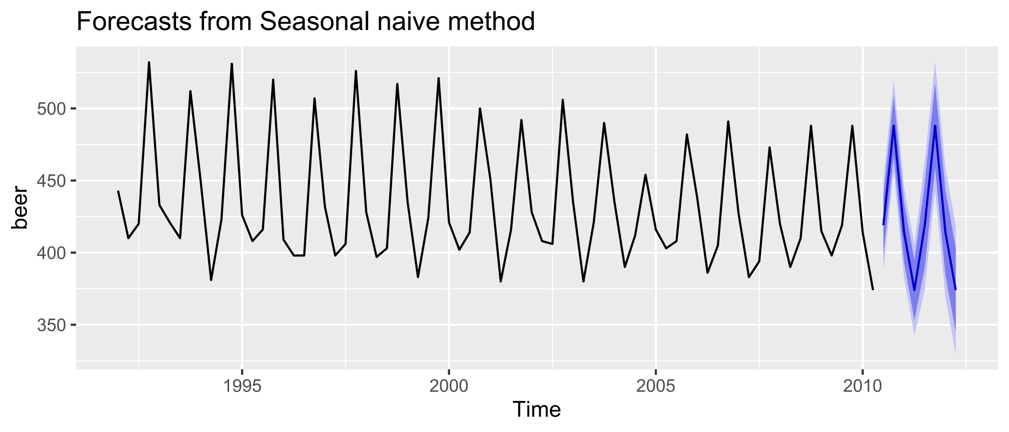


```
##
##      Ljung-Box test
##
## data:  Residuals from Naive method
## Q* = 11.031, df = 10, p-value = 0.3551
```

Your turn

Compute seasonal naïve forecasts for quarterly Australian beer production from 1992.

```
beer ← window(ausbeer, start=1992)
fc ← snaive(beer)
autoplot(fc)
```



Test if the residuals are white noise.

```
checkresiduals(fc)
```

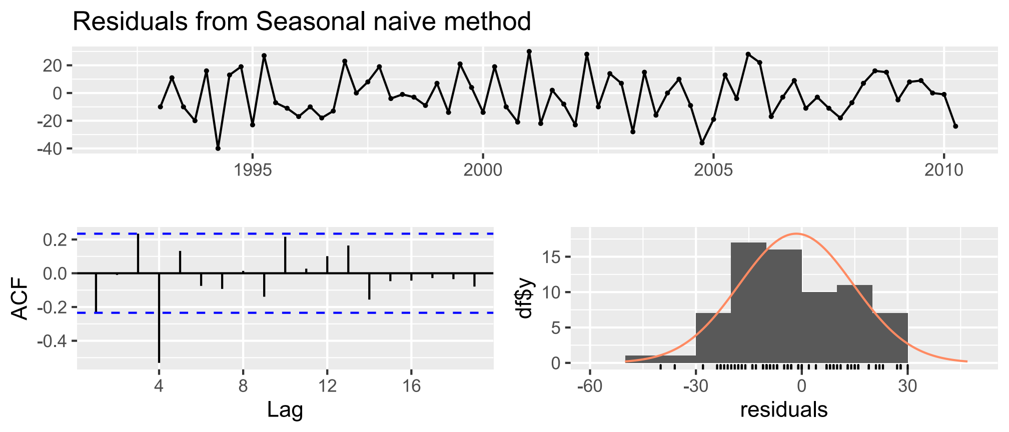
What do you conclude?

Sample solutions

Test if the residuals are white noise.

- **The results are statistically significant (i.e., the p -value< 0.05). We reject the null hypothesis H_0. Thus, we can conclude that the residuals are distinguishable from a white noise series**.

```
checkresiduals(fc)
```



```
##
##      Ljung-Box test
##
## data:  Residuals from Seasonal naive method
## Q* = 32.269, df = 8, p-value = 8.336e-05
```